arXiv:1502.02344v2 [stat.ML] 22 Jun 2015

Regularization Path of Cross-Validation Error Lower Bounds

Atsushi Shibagaki
Nagoya Institute of Technology
Nagoya, Japan
shibagaki.a.mllab.nit@gmail.com

Yoshiki Suzuki
Nagoya Institute of Technology
Nagoya, Japan
suzuki.mllab.nit@gmail.com

Masayuki Karasuyama
Nagoya Institute of Technology
Nagoya, Japan
karasuyama@nitech.ac.jp

Ichiro Takeuchi*
Nagoya Institute of Technology
Nagoya, Japan
takeuchi.ichiro@nitech.ac.jp

June 22, 2015

Abstract

Careful tuning of a *regularization parameter* is indispensable in many machine learning tasks because it has a significant impact on generalization performances. Nevertheless, current practice of regularization parameter tuning is more of an art than a science, e.g., it is hard to tell how many grid-points would be needed in cross-validation (CV) for obtaining a solution with sufficiently small CV error. In this paper we propose a novel framework for computing a lower bound of the CV errors as a function of the regularization parameter, which we call *regularization path of CV error lower bounds*. The proposed framework can be used for providing a theoretical approximation guarantee on a set of solutions in the sense that how far the CV error of the current best solution could be away from best possible CV error in the entire range of the regularization parameters. We demonstrate through numerical experiments that a theoretically guaranteed choice of a regularization parameter in the above sense is possible with reasonable computational costs.

1 Introduction

Many machine learning tasks involve careful tuning of *a regularization parameter* that controls the balance between an empirical loss term and a regularization term. A regularization parameter is usually selected

*Corresponding author

by comparing the cross-validation (CV) errors at several different regularization parameters. Although its choice has a significant impact on the generalization performances, the current practice is still more of an art than a science. For example, in commonly used grid-search, it is hard to tell how many grid points we should search over for obtaining sufficiently small CV error.

In this paper we introduce a novel framework for a class of regularized binary classification problems that can compute *a regularization path of CV error lower bounds*. For an $\varepsilon \in [0, 1]$, we define *ε-approximate regularization parameters* to be a set of regularization parameters such that the CV error of the solution at the regularization parameter is guaranteed to be no greater by ε than the best possible CV error in the entire range of regularization parameters. Given a set of solutions obtained, for example, by grid-search, the proposed framework allows us to provide a theoretical guarantee of the current best solution by explicitly quantifying its approximation level ε in the above sense. Furthermore, when a desired approximation level ε is specified, the proposed framework can be used for efficiently finding one of the ε-approximate regularization parameters.

The proposed framework is built on a novel CV error lower bound that can be represented as a function of the regularization parameter, and this is why we call it as a regularization path of CV error lower bounds. For computing a path, no special optimization algorithm is needed. We only need to have a finite number of solutions obtained by any algorithms. It is thus easy to apply our framework to common regularization parameter tuning strategies such as grid-search or *Bayesian optimization*. Furthermore, the proposed framework can be used not only with exact optimal solutions but also with sufficiently good approximate solutions, which is computationally advantageous because completely solving an optimization problem is often much more costly than obtaining a reasonably good approximate solution.

Our main contribution in this paper is to show that a theoretically guaranteed choice of a regularization parameter in the above sense is possible with reasonable computational costs. To the best of our knowledge, there is no other existing methods for providing such a theoretical guarantee on CV error that can be used as generally as ours. Figure 1 illustrates the behavior of the algorithm for obtaining $\varepsilon = 0.1$ approximate regularization parameter (see §5 for the setup).

Related works *Optimal regularization parameter* can be found if its *exact regularization path* can be computed. Exact regularization path has been intensively studied [8, 15], but they are known to be numerically unstable and do not scale well. Furthermore, exact regularization path can be computed only for a limited class of problems whose solutions are written as piecewise-linear functions of the regularization parameter [22]. Our framework is much more efficient and can be applied to wider classes of problems whose exact regularization path cannot be computed. This work was motivated by recent studies on approximate regularization path [13, 11, 12, 20]. These approximate regularization paths have a property that the objective function value at each regularization parameter value is no greater by ε than the optimal objective function value in the entire range of regularization parameters. Although these algorithms are much more stable and efficient than exact ones, for the task of tuning a regularization parameter, our interest is not in objective

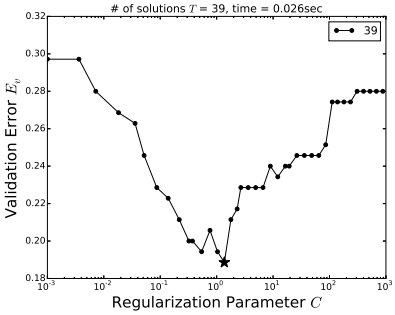


Figure 1: An illustration of the proposed framework. One of our algorithms presented in §4 automatically selected 39 regularization parameter values in $[10^{-3}, 10^3]$, and an upper bound of the validation error for each of them is obtained by solving an optimization problem approximately. Among those 39 values, the one with the smallest validation error upper bound (indicated as ★ at $C = 1.368$) is guaranteed to be $\varepsilon(= 0.1)$ *approximate regularization parameter* in the sense that the validation error for the regularization parameter is no greater by ε than the smallest possible validation error in the whole interval $[10^{-3}, 10^3]$. See §5 for the setup (see also Figure 3 for the results with other options).

function values but in CV errors. Our approach is more suitable for regularization parameter tuning tasks in the sense that the approximation quality is guaranteed in terms of CV error.

As illustrated in Figure 1, we only compute a finite number of solutions, but still provide approximation guarantee in the whole interval of the regularization parameter. To ensure such a property, we need to introduce a novel CV error lower bound that is sufficiently tight and represented as a monotonic function of the regularization parameter. Although several CV error bounds (mostly for leave-one-out CV) of SVM and other similar learning frameworks exist (e.g., [26, 16, 7, 17]), none of them satisfy the above required properties. The idea of our CV error bound is inspired from recent studies on *safe screening* [9, 28, 21, 19, 27] (see Appendix A for the detail). Furthermore, we emphasize that our contribution is *not* in presenting a new generalization error bound, but in introducing a practical framework for providing a theoretical guarantee on the choice of a regularization parameter. Although generalization error bounds such as structural risk minimization [25] might be used for a rough tuning of a regularization parameter, they are known to be too loose to use as an alternative to CV (see, e.g., §11 in [23]). We also note that our contribution is *not* in presenting new method for regularization parameter tuning such as Bayesian optimization [24], random search [1] and gradient-based search [6]. As we demonstrate in experiments, our approach can provide a theoretical approximation guarantee of the regularization parameter selected by these existing methods.

2 Problem Setup

We consider linear binary classification problems. Let $\{(x_i, y_i) \in \mathbb{R}^d \times \{-1, 1\}\}_{i \in [n]}$ be the training set where n is the size of the training set, d is the input dimension, and $[n] := \{1, \ldots, n\}$. An independent held-out validation set with size n' is denoted similarly as $\{(x'_i, y'_i) \in \mathbb{R}^d \times \{-1, 1\}\}_{i \in [n']}$. A linear decision function is written as $f(x) = w^\top x$, where $w \in \mathbb{R}^d$ is a vector of coefficients, and $^\top$ represents the transpose. We assume the availability of a held-out validation set only for simplifying the exposition. All the proposed

methods presented in this paper can be straightforwardly adapted to a cross-validation setup. Furthermore, the proposed methods can be *kernelized* if the loss function satisfies a certain condition. In this paper we focus on the following class of regularized convex loss minimization problems:

$$w_C^* := \arg\min_{w\in\mathbb{R}^d} \frac{1}{2}\|w\|^2 + C\sum_{i\in[n]} \ell(y_i, w^\top x_i), \tag{1}$$

where $C > 0$ is the regularization parameter, and $\|\cdot\|$ is the Euclidean norm. The loss function is denoted as $\ell : \{-1, 1\} \times \mathbb{R} \to \mathbb{R}$. We assume that $\ell(\cdot, \cdot)$ is convex and subdifferentiable in the 2nd argument. Examples of such loss functions include logistic loss, hinge loss, Huber-hinge loss, etc. For notational convenience, we denote the individual loss as $\ell_i(w) := \ell(y_i, w^\top x_i)$ for all $i \in [n]$. The optimal solution for the regularization parameter C is explicitly denoted as w_C^*. We assume that the regularization parameter is defined in a finite interval $[C_\ell, C_u]$, e.g., $C_\ell = 10^{-3}$ and $C_u = 10^3$ as we did in the experiments.

For a solution $w \in \mathbb{R}^d$, the validation error[1] is defined as

$$E_v(w) := \frac{1}{n'}\sum_{i\in[n']} I(y_i' w^\top x_i' < 0), \tag{2}$$

where $I(\cdot)$ is the indicator function. In this paper, we consider two problems. In the first problem, given a set of (either optimal or approximate) solutions $w_{C_1}^*, \ldots, w_{C_T}^*$ at T different regularization parameters $C_1, \ldots, C_T \in [C_\ell, C_u]$, we compute the approximation level ε such that

$$\min_{C_t\in\{C_1,\ldots,C_T\}} E_v(w_{C_t}^*) - E_v^* \le \varepsilon, \quad \text{where} \quad E_v^* := \min_{C\in[C_l,C_u]} E_v(w_C^*). \tag{3}$$

In the second problem, we find an ε-approximate regularization parameter within an interval $C \in [C_l, C_u]$, which is defined as an element of the following set

$$\mathcal{C}(\varepsilon) := \Big\{ C \in [C_l, C_u] \;\Big|\; E_v(w_C^*) - E_v^* \le \varepsilon \Big\}.$$

Both of these two problems can be solved by using our proposed framework for computing a path of validation error lower bounds.

3 Validation error lower bounds as a function of regularization parameter

In this section, we derive a validation error lower bound which is represented as a function of the regularization parameter C. Our basic idea is to compute a lower and an upper bound of the inner product score $w_C^{*\top} x_i'$ for each validation input $x_i', i \in [n']$, as a function of the regularization parameter C. For computing the bounds of $w_C^{*\top} x_i'$, we use a solution (either optimal or approximate) for a different regularization parameter $\tilde{C} \neq C$.

[1] For simplicity, we regard a validation instance whose score is exactly zero, i.e., $w^\top x_i' = 0$, is correctly classified in (2). Hereafter, we assume that there are no validation instances whose input vector is completely 0, i.e., $x_i' = 0$, because those instances are always correctly classified according to the definition in (2).

3.1 Score bounds

We first describe how to obtain a lower and an upper bound of inner product score $w_C^{*\top} x'_i$ based on an approximate solution $\hat{w}_{\tilde{C}}$ at a different regularization parameter $\tilde{C} \neq C$.

Lemma 1. *Let $\hat{w}_{\tilde{C}}$ be an approximate solution of the problem (1) for a regularization parameter value $\tilde{C}$ and $\xi_i(\hat{w}_C)$ be a subgradient of ℓ_i at $w = \hat{w}_C$ such that a subgradient of the objective function is*

$$g(\hat{w}_{\tilde{C}}) := \hat{w}_C + \tilde{C} \sum_{i \in [n]} \xi_i(\hat{w}_C). \tag{4}$$

Then, for any $C > 0$, the score $w_C^{*\top} x'_i, i \in [n']$, satisfies

$$w_C^{*\top} x'_i \geq LB(w_C^{*\top} x'_i | \hat{w}_{\tilde{C}}) := \begin{cases} \alpha(\hat{w}_{\tilde{C}}, x'_i) - \frac{1}{\tilde{C}}(\beta(\hat{w}_{\tilde{C}}, x'_i) + \gamma(g(\hat{w}_{\tilde{C}}), x'_i))C, \text{if } C > \tilde{C}, \\ -\beta(\hat{w}_{\tilde{C}}, x'_i) + \frac{1}{\tilde{C}}(\alpha(\hat{w}_{\tilde{C}}, x'_i) + \delta(g(\hat{w}_{\tilde{C}}), x'_i))C, \text{if } C < \tilde{C}, \end{cases} \tag{5a}$$

$$w_C^{*\top} x'_i \leq UB(w_C^{*\top} x'_i | \hat{w}_{\tilde{C}}) := \begin{cases} -\beta(\hat{w}_{\tilde{C}}, x'_i) + \frac{1}{\tilde{C}}(\alpha(\hat{w}_{\tilde{C}}, x'_i) + \delta(g(\hat{w}_{\tilde{C}}), x'_i))C, \text{if } C > \tilde{C}, \\ \alpha(\hat{w}_{\tilde{C}}, x'_i) - \frac{1}{\tilde{C}}(\beta(\hat{w}_{\tilde{C}}, x'_i) + \gamma(g(\hat{w}_{\tilde{C}}), x'_i))C, \text{if } C < \tilde{C}, \end{cases} \tag{5b}$$

where

$$\alpha(w^*_{\tilde{C}}, x'_i) := \frac{1}{2}(\|w^*_{\tilde{C}}\|\|x'_i\| + w^{*\top}_{\tilde{C}} x'_i) \geq 0,\ \gamma(g(\hat{w}_{\tilde{C}}), x'_i) := \frac{1}{2}(\|g(\hat{w}_{\tilde{C}})\|\|x'_i\| + g(\hat{w}_{\tilde{C}})^\top x'_i) \geq 0,$$
$$\beta(w^*_{\tilde{C}}, x'_i) := \frac{1}{2}(\|w^*_{\tilde{C}}\|\|x'_i\| - w^{*\top}_{\tilde{C}} x'_i) \geq 0,\ \delta(g(\hat{w}_{\tilde{C}}), x'_i) := \frac{1}{2}(\|g(\hat{w}_{\tilde{C}})\|\|x'_i\| - g(w_{\tilde{C}})^\top x'_i) \geq 0.$$

The proof is presented in Appendix A. Lemma 1 tells that we have a lower and an upper bound of the score $w_C^{*\top} x'_i$ for each validation instance that linearly change with the regularization parameter C. When $\hat{w}_{\tilde{C}}$ is optimal, it can be shown that (see Proposition B.24 in [2]) there exists a subgradient such that $g(\hat{w}_{\tilde{C}}) = 0$, meaning that the bounds are tight because $\gamma(g(\hat{w}_{\tilde{C}}), x'_i) = \delta(g(\hat{w}_{\tilde{C}}), x'_i) = 0$.

Corollary 2. *When $C = \tilde{C}$, the score $w_{\tilde{C}}^{*\top} x'_i, i \in [n']$, for the regularization parameter value $\tilde{C}$ itself satisfies*

$$w_{\tilde{C}}^{*\top} x'_i \geq LB(w_{\tilde{C}}^{*\top} x'_i | \hat{w}_{\tilde{C}}) = \hat{w}_{\tilde{C}}^\top x'_i - \gamma(g(\hat{w}_{\tilde{C}}), x'_i),\ w_{\tilde{C}}^{*\top} x'_i \leq UB(w_{\tilde{C}}^{*\top} x'_i | \hat{w}_{\tilde{C}}) = \hat{w}_{\tilde{C}}^\top x'_i + \delta(g(\hat{w}_{\tilde{C}}), x'_i).$$

The results in Corollary 2 are obtained by simply substituting $C = \tilde{C}$ into (5a) and (5b).

3.2 Validation Error Bounds

Given a lower and an upper bound of the score of each validation instance, a lower bound of the validation error can be computed by simply using the following facts:

$$y'_i = +1 \text{ and } UB(w_C^{*\top} x'_i | \hat{w}_{\tilde{C}}) < 0 \Rightarrow \text{mis-classified}, \tag{6a}$$

$$y'_i = -1 \text{ and } LB(w_C^{*\top} x'_i | \hat{w}_{\tilde{C}}) > 0 \Rightarrow \text{mis-classified}. \tag{6b}$$

Furthermore, since the bounds in Lemma 1 linearly change with the regularization parameter C, we can identify the interval of C within which the validation instance is guaranteed to be mis-classified.

Lemma 3. *For a validation instance with $y'_i = +1$, if*

$$\tilde{C} < C < \frac{\beta(\hat{w}_{\tilde{C}}, x'_i)}{\alpha(\hat{w}_{\tilde{C}}, x'_i) + \delta(g(\hat{w}_{\tilde{C}}), x'_i)}\tilde{C} \quad \text{or} \quad \frac{\alpha(\hat{w}_{\tilde{C}}, x'_i)}{\beta(\hat{w}_{\tilde{C}}, x'_i) + \gamma(g(\hat{w}_{\tilde{C}}), x'_i)}\tilde{C} < C < \tilde{C},$$

then the validation instance (x'_i, y'_i) is mis-classified. Similarly, for a validation instance with $y'_i = -1$, if

$$\tilde{C} < C < \frac{\alpha(\hat{w}_{\tilde{C}}, x'_i)}{\beta(\hat{w}_{\tilde{C}}, x'_i) + \gamma(g(\hat{w}_{\tilde{C}}), x'_i)}\tilde{C} \quad \text{or} \quad \frac{\beta(\hat{w}_{\tilde{C}}, x'_i)}{\alpha(\hat{w}_{\tilde{C}}, x'_i) + \delta(g(\hat{w}_{\tilde{C}}), x'_i)}\tilde{C} < C < \tilde{C},$$

then the validation instance (x'_i, y'_i) is mis-classified.

This lemma can be easily shown by applying (5) to (6).

Using Lemma 3, the lower bound of the validation error is represented as a function of the regularization parameter C in the following form.

Theorem 4. *Using an approximate solution $\hat{w}_{\tilde{C}}$ for a regularization parameter $\tilde{C}$, the validation error $E_v(w^*_C)$ for any $C > 0$ satisfies*

$$E_v(w^*_C) \geq LB(E_v(w^*_C)|\hat{w}_{\tilde{C}}) := \tag{7}$$
$$\frac{1}{n'}\Bigg(\sum_{y'_i=+1} I\Bigg(\tilde{C}<C<\frac{\beta(\hat{w}_{\tilde{C}}, x'_i)}{\alpha(\hat{w}_{\tilde{C}}, x'_i)+\delta(g(\hat{w}_{\tilde{C}}), x'_i)}\tilde{C}\Bigg)+\sum_{y'_i=+1} I\Bigg(\frac{\alpha(\hat{w}_{\tilde{C}}, x'_i)}{\beta(\hat{w}_{\tilde{C}}, x'_i)+\gamma(g(\hat{w}_{\tilde{C}}), x'_i)}\tilde{C}<C<\tilde{C}\Bigg)$$
$$+\sum_{y'_i=-1} I\Bigg(\tilde{C}<C<\frac{\alpha(\hat{w}_{\tilde{C}}, x'_i)}{\beta(\hat{w}_{\tilde{C}}, x'_i)+\gamma(g(\hat{w}_{\tilde{C}}), x'_i)}\tilde{C}\Bigg)+\sum_{y'_i=-1} I\Bigg(\frac{\beta(\hat{w}_{\tilde{C}}, x'_i)}{\alpha(\hat{w}_{\tilde{C}}, x'_i)+\delta(g(\hat{w}_{\tilde{C}}), x'_i)}\tilde{C}<C<\tilde{C}\Bigg)\Bigg).$$

Theorem 4 is a direct consequence of Lemma 3. The lower bound (7) is a staircase function of the regularization parameter C.

By setting $C = \tilde{C}$, we can obtain a lower and an upper bound of the validation error for the regularization parameter $\tilde{C}$ itself, which are used in the algorithm as a stopping criteria for obtaining an approximate solution $\hat{w}_{\tilde{C}}$.

Corollary 5. *Given an approximate solution $\hat{w}_{\tilde{C}}$, the validation error $E_v(w^*_{\tilde{C}})$ satisfies*

$$E_v(w^*_{\tilde{C}}) \geq LB(E_v(w^*_{\tilde{C}})|\hat{w}_{\tilde{C}})$$
$$= \frac{1}{n'}\Bigg(\sum_{y'_i=+1} I\big(\hat{w}_{\tilde{C}}^\top x'_i + \delta(g(\hat{w}_{\tilde{C}}), x'_i) < 0\big) + \sum_{y'_i=-1} I\big(\hat{w}_{\tilde{C}}^\top x'_i - \gamma(g(\hat{w}_{\tilde{C}}), x'_i) > 0\big)\Bigg), \tag{8a}$$
$$E_v(w^*_{\tilde{C}}) \leq UB(E_v(w^*_{\tilde{C}})|\hat{w}_{\tilde{C}})$$
$$= 1 - \frac{1}{n'}\Bigg(\sum_{y'_i=+1} I\big(\hat{w}_{\tilde{C}}^\top x'_i - \gamma(g(\hat{w}_{\tilde{C}}), x'_i) \geq 0\big) + \sum_{y'_i=-1} I\big(\hat{w}_{\tilde{C}}^\top x'_i + \delta(g(\hat{w}_{\tilde{C}}), x'_i) \leq 0\big)\Bigg). \tag{8b}$$

algorithm

4 Algorithm

In this section we present two algorithms for each of the two problems discussed in §2. Due to the space limitation, we roughly describe the most fundamental forms of these algorithms. Details and several extensions

Algorithm 1: Computing the approximation level ε from the given set of solutions

Input: $\{(x_i, y_i)\}_{i\in[n]}$, $\{(x'_i, y'_i)\}_{i\in[n']}$, C_l, C_u, $\mathcal{W} := \{w_{\tilde{C}_1}, \ldots, w_{\tilde{C}_T}\}$

1: $E_v^{\text{best}} \leftarrow \min_{\tilde{C}_t\in\{\tilde{C}_1,\ldots,\tilde{C}_T\}} UB(E_v(w^*_{\tilde{C}_t})|w_{\tilde{C}_t})$

2: $LB(E_v^*) \leftarrow \min_{c\in[C_l,C_u]} \left\{ \max_{\tilde{C}_t\in\{\tilde{C}_1,\ldots,\tilde{C}_T\}} LB(E_v(w_c^*)|w_{\tilde{C}_t})\right\}$

Output: $\varepsilon = E_v^{\text{best}} - LB(E_v^*)$

Algorithm 2: Finding an ε approximate regularization parameter with approximate solutions

Input: $\{(x_i, y_i)\}_{i\in[n]}$, $\{(x'_i, y'_i)\}_{i\in[n']}$, C_l, C_u, ε

1: $t \leftarrow 1$, $\tilde{C}_t \leftarrow C_l$, $C^{\text{best}} \leftarrow C_l$, $E_v^{\text{best}} \leftarrow 1$

2: **while** $\tilde{C}_t \le C_u$ **do**

3: $\quad\hat{w}_{\tilde{C}_t} \leftarrow$ solve (1) approximately for $C = \tilde{C}_t$

4: $\quad$Compute $UB(E_v(w^*_{\tilde{C}_t})|\hat{w}_{\tilde{C}_t})$ by (8b).

5: $\quad$**if** $UB(E_v(w^*_{\tilde{C}_t})|\hat{w}_{\tilde{C}_t}) < E_v^{\text{best}}$ **then**

6: $\quad\quad E_v^{\text{best}} \leftarrow UB(E_v(w^*_{\tilde{C}_t})|\hat{w}_{\tilde{C}_t})$, $C^{\text{best}} \leftarrow \tilde{C}_t$

7: $\quad$**end if**

8: $\quad$Set $\tilde{C}_{t+1}$ by (10)

9: $\quad t \leftarrow t + 1$

10: **end while**

Output: $C^{\text{best}} \in \mathcal{C}(\varepsilon)$.

of the algorithms are presented in supplementary appendices B , C and D.

4.1 Problem 1: Computing the approximation level ε from a given set of solutions

Given a set of (either optimal or approximate) solutions $\hat{w}_{\tilde{C}_1}, \ldots, \hat{w}_{\tilde{C}_T}$, obtained e.g., by ordinary grid-search, our first problem is to provide a theoretical approximation level ε in the sense of (3)[2]. This problem can be solved easily by using the validation error lower bounds developed in §3.2. The algorithm is presented in Algorithm 1, where we compute the current best validation error E_v^{best} in line 1, and a lower bound of the best possible validation error $E_v^* := \min_{C\in[C_\ell,C_u]} E_v(w_C^*)$ in line 2. Then, the approximation level ε can be simply obtained by subtracting the latter from the former. We note that $LB(E_v^*)$, the lower bound of E_v^*, can be easily computed by using T valuation error lower bounds $LB(E_v(w_C^*)|w_{\tilde{C}_t})$, $t = 1, \ldots, T$, because they are represented as staircase functions of C.

[2] When we only have approximate solutions $\hat{w}_{\tilde{C}_1}, \ldots, \hat{w}_{\tilde{C}_T}$, Eq. (3) is slightly incorrect. The first term of the l.h.s. of (3) should be $\min_{\tilde{C}_t\in\{\tilde{C}_1,\ldots,\tilde{C}_T\}} UB(E_v(\hat{w}_{\tilde{C}_t})|\hat{w}_{\tilde{C}_t})$.

4.2 Problem 2: Finding an ε-approximate regularization parameter

Given a desired approximation level ε such as $\varepsilon = 0.01$, our second problem is to find an ε-approximate regularization parameter. To this end we develop an algorithm that produces a set of optimal or approximate soluitons $\hat{w}_{\tilde{C}_1}, \ldots, \hat{w}_{\tilde{C}_T}$ such that, if we apply Algorithm 1 to this sequence, then approximation level would be smaller than or equal to ε. Algorithm 2 is the pseudo-code of this algorithm. It computes approximate solutions for an increasing sequence of regularization parameters in the main loop (lines 2-11).

Let us now consider t^{th} iteration in the main loop, where we have already computed $t-1$ approximate solutions $\hat{w}_{\tilde{C}_1}, \ldots, \hat{w}_{\tilde{C}_{t-1}}$ for $\tilde{C}_1 < \ldots < \tilde{C}_{t-1}$. At this point,

$$C^{\text{best}} := \arg \min_{\tilde{C}_\tau \in \{\tilde{C}_1, \ldots, \tilde{C}_{t-1}\}} UB(E_v(w^*_{\tilde{C}_\tau})|\hat{w}_{\tilde{C}_\tau}),$$

is the best (in worst-case) regularization parameter obtained so far and it is guaranteed to be an ε-approximate regularization parameter in the interval $[C_l, \tilde{C}_t]$ in the sense that the validation error,

$$E_v^{\text{best}} := \min_{\tilde{C}_\tau \in \{\tilde{C}_1, \ldots, \tilde{C}_{t-1}\}} UB(E_v(w^*_{\tilde{C}_\tau})|\hat{w}_{\tilde{C}_\tau}),$$

is shown to be at most greater by ε than the smallest possible validation error in the interval $[C_l, \tilde{C}_t]$. However, we are not sure whether C^{best} can still keep ε-approximation property for $C > \tilde{C}_t$. Thus, in line 3, we approximately solve the optimization problem (1) at $C = \tilde{C}_t$ and obtain an approximate solution $\hat{w}_{\tilde{C}_t}$. Note that the approximate solution $\hat{w}_{\tilde{C}_t}$ must be sufficiently good enough in the sense that $UB(E_v(w^*_{\tilde{C}_\tau})|\hat{w}_{\tilde{C}_\tau}) - LB(E_v(w^*_{\tilde{C}_\tau})|\hat{w}_{\tilde{C}_\tau})$ is sufficiently smaller than ε (typically 0.1ε). If the upper bound of the validation error $UB(E_v(w^*_{\tilde{C}_\tau})|\hat{w}_{\tilde{C}_\tau})$ is smaller than E_v^{best}, we update E_v^{best} and C^{best} (lines 5-8).

Our next task is to find $\tilde{C}_{t+1}$ in such a way that C^{best} is an ε-approximate regularization parameter in the interval $[C_l, \tilde{C}_{t+1}]$. Using the validation error lower bound in Theorem 4, the task is to find the smallest $\tilde{C}_{t+1} > \tilde{C}_t$ that violates

$$E_v^{\text{best}} - LB(E_v(w^*_C)|\hat{w}_{\tilde{C}_t}) \leq \varepsilon, \quad \forall C \in [\tilde{C}_t, C_u], \tag{9}$$

In order to formulate such a $\tilde{C}_{t+1}$, let us define

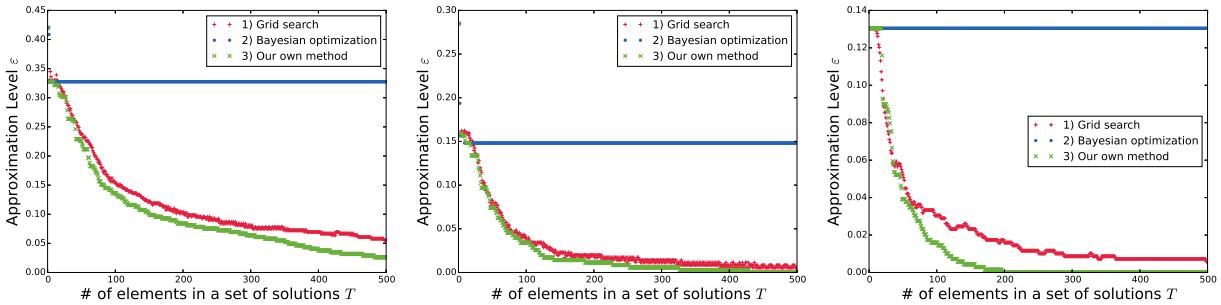
$$\mathcal{P} := \{i \in [n'] | y'_i = +1, UB(w^{*\top}_{\tilde{C}_t} x'_i | \hat{w}_{\tilde{C}_t}) < 0\}, \mathcal{N} := \{i \in [n'] | y'_i = -1, LB(w^{*\top}_{\tilde{C}_t} x'_i | \hat{w}_{\tilde{C}_t}) > 0\}.$$

Furthermore, let

$$\Gamma := \left\{ \frac{\beta(\hat{w}_{\tilde{C}_t}, x'_i)}{\alpha(\hat{w}_{\tilde{C}_t}, x'_i) + \delta(g(\hat{w}_{\tilde{C}_t}), x'_i)} \tilde{C}_t \right\}_{i \in \mathcal{P}} \cup \left\{ \frac{\alpha(\hat{w}_{\tilde{C}_t}, x'_i)}{\beta(\hat{w}_{\tilde{C}_t}, x'_i) + \gamma(g(\hat{w}_{\tilde{C}_t}), x'_i)} \tilde{C}_t \right\}_{i \in \mathcal{N}},$$

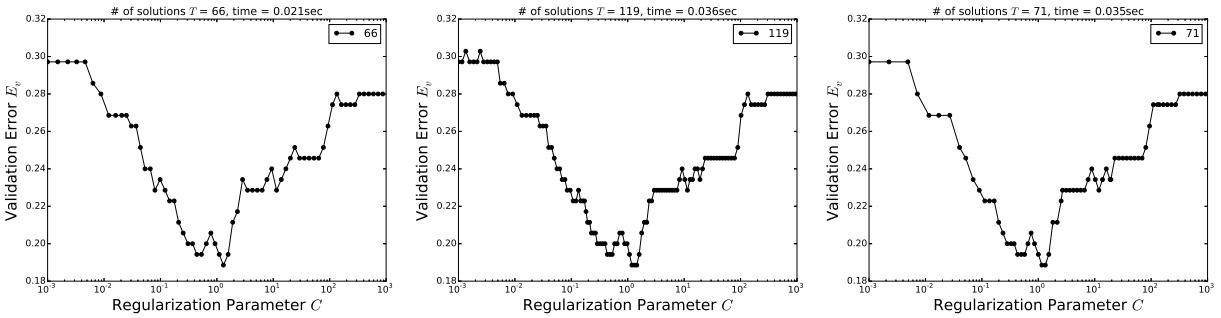
and denote the k^{th}-smallest element of Γ as $k^{\text{th}}(\Gamma)$ for any natural number k. Then, the smallest $\tilde{C}_{t+1} > \tilde{C}_t$ that violates (9) is given as

$$\tilde{C}_{t+1} \leftarrow (\lfloor n'(LB(E_v(w^*_{\tilde{C}_t})|\hat{w}_{\tilde{C}_t}) - E_v^{\text{best}} + \varepsilon) \rfloor + 1)^{\text{th}}(\Gamma). \tag{10}$$



`liver-disorders` (D2) `ionosphere` (D3) `australian` (D4)

Figure 2: Illustrations of Algorithm 1 on three benchmark datasets (D2, D3, D4). The plots indicate how the approximation level ε improves as the number of solutions T increases in grid-search (red), Bayesian optimization (blue) and our own method (green, see the main text).



(a) $\varepsilon = 0.1$ without tricks (b) $\varepsilon = 0.05$ without tricks (c) $\varepsilon = 0.05$ with tricks 1 and 2

Figure 3: Illustrations of Algorithm 2 on `ionosphere` (D3) dataset for (a) **op2** with $\varepsilon = 0.10$, (b) **op2** with $\varepsilon = 0.05$ and (c) **op3** with $\varepsilon = 0.05$, respectively. Figure 1 also shows the result for **op3** with $\varepsilon = 0.10$.

5 Experiments

In this section we present experiments for illustrating the proposed methods. Table 2 summarizes the datasets used in the experiments. They are taken from libsvm dataset repository [4]. All the input features except D9 and D10 were standardized to $[-1, 1]$[3]. For illustrative results, the instances were randomly divided into a training and a validation sets in roughly equal sizes. For quantitative results, we used 10-fold CV. We used Huber hinge loss (e.g., [5]) which is convex and subdifferentiable with respect to the second argument. The proposed methods are free from the choice of optimization solvers. In the experiments, we used an optimization solver described in [18], which is also implemented in well-known *liblinear* software [10]. Our slightly modified code (for adaptation to Huber hinge loss) is provided as a supplementary material, and it will be put in public domain after the paper is accepted. Whenever possible, we used *warm-start* approach,

[3] We use D9 and D10 as they are for exploiting sparsity.

i.e., when we trained a new solution, we used the closest solutions trained so far (either approximate or optimal ones) as the initial starting point of the optimizer. All the computations were conducted by using a single core of an HP workstation Z800 (Xeon(R) CPU X5675 (3.07GHz), 48GB MEM). In all the experiments, we set $C_\ell = 10^{-3}$ and $C_u = 10^3$.

Results on problem 1 We applied Algorithm 1 in §4 to a set of solutions obtained by 1) grid-search, 2) Bayesian optimization (with expected improvement acquisition function), and 3) our own method that exploits information on the CV error lower bound available during the search process. Figure 2 illustrates the results on three datasets, where we see how the approximation level ε in the vertical axis changes as the number of solutions (T in our notation) increases. The red plots indicate the results of grid-search. As we increase the grid points, the approximation level ε was tended to be improved. The blue plots indicate the results of Bayesian Optimization (BO). Since BO tends to focus its search on a small region of the regularization parameter, it was difficult to tightly bound the approximation level. The green plots indicate the result of the third option, where we sequentially computed a solution whose validation error lower bound is smallest based on the information obtained so far. The results suggest that this naive approach seems to offer slight improvement from grid-search.

Results on problem 2 We applied Algorithm 2 to benchmark datasets for demonstrating theoretically guaranteed choice of a regularization parameter is possible with reasonable computational costs. Besides the algorithm presented in §4, we also tested a variant described in supplementary Appendix B. Specifically, we have three algorithm options. In the first option (**op1**), we used optimal solutions $\{w^*_{\tilde{C}_t}\}_{t\in[T]}$ for computing CV error lower bounds. In the second option (**op2**), we instead used approximate solutions $\{\hat{w}_{\tilde{C}_t}\}_{t\in[T]}$. In the last option (**op3**), we additionally used speed-up tricks described in supplementary Appendix B. We considered four different choices of $\varepsilon \in \{0.1, 0.05, 0.01, 0\}$. Note that $\varepsilon = 0$ indicates the task of finding the exactly optimal regularization parameter. In some datasets, the smallest validation errors are less than 0.1 or 0.05, in which cases we do not report the results (indicated as "$E_v < 0.05$" etc.). In trick1, we initially computed solutions at four different regularization parameter values evenly allocated in $[10^{-3}, 10^3]$ in the logarithmic scale. In trick2, the next regularization parameter $\tilde{C}_{t+1}$ was set by replacing ε in (10) with 1.5ε (see supplementary Appendix B).

For the purpose of illustration, we plot examples of validation error curves in several setups. Figure 3 shows the validation error curves of `ionosphere` (D3) dataset for several options and ε.

Next, we report the results on computational costs in CV setups. Table 1 shows the number of optimization problems we actually solved in the algorithm (which is denoted as T), and the total computation time in seconds. The computational costs of the methods mostly depend on T. As is evident from the algorithm description in §4, T gets smaller as ε increases. Two tricks in supplementary Appendix B seem to be helpful in most cases for reducing T. In addition, we see the advantage of using approximate solutions by comparing the computation times of **op1** and **op2**, although approximate solutions can be only used for $\varepsilon \neq 0$. Over-

Table 1: Computational costs. For each of the three options and $\varepsilon \in \{0.10, 0.05, 0.01, 0\}$, the number of optimization problems solved (denoted as T) and the total computational costs (denoted as time) are listed. Note that, for **op2**, there are no results for $\varepsilon = 0$.

		op1 (using $w^*_{\bar{C}}$)		**op2** (using $\hat{w}_{\bar{C}}$)		**op3** (using tricks)			**op1** (using $w^*_{\bar{C}}$)		**op2** (using $\hat{w}_{\bar{C}}$)		**op3** (using tricks)	
ε		T	time (sec)	T	time (sec)	T	time (sec)		T	time (sec)	T	time (sec)	T	time (sec)
0.10	D1	30	0.068	32	0.031	33	0.041	D6	92	1.916	93	0.975	62	0.628
0.05		68	0.124	70	0.061	57	0.057		207	4.099	209	2.065	123	1.136
0.01		234	0.428	324	0.194	205	0.157		1042	16.31	1069	9.686	728	5.362
0		442	0.697	N.A.		383	0.629		4276	57.57	N.A.		2840	44.68
0.10	D2	221	0.177	223	0.124	131	0.084	D7	289	8.492	293	5.278	167	3.319
0.05		534	0.385	540	0.290	367	0.218		601	16.18	605	9.806	379	6.604
0.01		1503	0.916	2183	0.825	1239	0.623		2532	57.79	2788	35.21	1735	24.04
0		10939	6.387	N.A.		6275	3.805		67490	1135	N.A.		42135	760.8
0.10	D3	61	0.617	62	0.266	43	0.277	D8	72	0.761	74	0.604	66	0.606
0.05		123	1.073	129	0.468	73	0.359		192	1.687	195	1.162	110	0.926
0.01		600	4.776	778	0.716	270	0.940		1063	8.257	1065	6.238	614	4.043
0		5412	26.39	N.A.		815	6.344		34920	218.4	N.A.		15218	99.57
0.10	D4	27	0.169	27	0.088	23	0.093	D9	134	360.2	136	201.0	89	74.37
0.05		64	0.342	65	0.173	47	0.153		317	569.9	323	280.7	200	128.5
0.01		167	0.786	181	0.418	156	0.399		1791	2901	1822	1345	1164	657.4
0		342	1.317	N.A.		345	1.205		85427	106937	N.A.		63300	98631
0.10	D5	62	0.236	63	0.108	45	0.091	D10	$E_v < 0.10$		$E_v < 0.10$		$E_v < 0.10$	
0.05		108	0.417	109	0.171	77	0.137		$E_v < 0.05$		$E_v < 0.05$		$E_v < 0.05$	
0.01		421	1.201	440	0.631	258	0.401		157	81.75	162	31.02	114	36.81
0		2330	4.540	N.A.		968	2.451		258552	85610	N.A.		42040	23316

all, the results suggest that the proposed algorithm allows us to find theoretically guaranteed approximate regularization parameters with reasonable costs except for $\varepsilon = 0$ cases. For example, the algorithm found an $\varepsilon = 0.01$ approximate regularization parameter within a minute in 10-fold CV for a dataset with more than 50000 instances (see the results on D10 for $\varepsilon = 0.01$ with **op2** and **op3** in Table 1).

Table 2: Benchmark datasets used in the experiments.

	dataset name	sample size	input dimension		dataset name	sample size	input dimension
D1	`heart`	270	13	D6	`german.numer`	1000	24
D2	`liver-disorders`	345	6	D7	`svmguide3`	1284	21
D3	`ionosphere`	351	34	D8	`svmguide1`	7089	4
D4	`australian`	690	14	D9	`a1a`	32561	123
D5	`diabetes`	768	8	D10	`w8a`	64700	300

6 Conclusions and future works

We presented a novel algorithmic framework for computing CV error lower bounds as a function of the regularization parameter. The proposed framework can be used for a theoretically guaranteed choice of a regularization parameter. Additional advantage of this framework is that we only need to compute a set of sufficiently good approximate solutions for obtaining such a theoretical guarantee, which is computationally

advantageous. As demonstrated in the experiments, our algorithm is practical in the sense that the computational cost is reasonable as long as the approximation quality ε is not too close to 0. An important future work is to extend the approach to multiple hyper-parameters tuning setups.

A Proof of Lemma 1

In this section we prove Lemma 1. First we present two propositions which are used of proving Lemma 1.

Proposition 6. *Consider the following general problem:*

$$\min_{z} \ \phi(z) \quad \text{s.t. } z \in \mathcal{Z}, \tag{11}$$

where $\phi : \mathcal{Z} \to \mathbb{R}$ is a subdifferentiable convex function and $\mathcal{Z} \subset \mathbb{R}^d$ is a convex set. Then a solution z^* is the optimal solution of (11) if and only if there exists a subgradient $\xi \in \partial\phi(z^*)$ such that

$$\xi^\top (z^* - z) \leq 0, \quad \forall \ z \in \mathcal{Z},$$

where $\partial\phi(z^*)$ is the set of all subgradients of convex function ϕ at $z = z^*$.

See, for example, Proposition B.24 in [2] for the proof of Proposition 6.

Proposition 7. *Let $p, q \in \mathbb{R}^d$ be arbitrary d-dimensional vectors and $r > 0$ be an arbitrary positive constant. Then, the solutions of the following optimization problem can be explicitly obtained as follows:*

$$p^\top q - \|p\| r = \min_{z \in \mathbb{R}^d} \ p^\top z \quad \text{s.t. } \|z - q\|^2 \leq r^2, \tag{12}$$

$$p^\top q + \|p\| r = \max_{z \in \mathbb{R}^d} \ p^\top z \quad \text{s.t. } \|z - q\|^2 \leq r^2. \tag{13}$$

Proof of Proposition 7. Using a Lagrange multiplier $\lambda > 0$, the problem (12) is rewritten as

$$\begin{aligned}
&\min_{z \in \mathbb{R}^d} \ p^\top z \quad \text{s.t. } \|z - q\|^2 \leq r^2 \\
=&\min_{z \in \mathbb{R}^d} \max_{\lambda > 0} \big(p^\top z + \lambda (\|z - q\|^2 \leq r^2) \big) \\
=&\max_{\lambda > 0} \big(-\lambda r^2 + \min_{z} \big(\lambda \|z - p\|^2 + p^\top z \big) \big) \\
=&\max_{\lambda > 0} \ H(\lambda) := \big(-\lambda r^2 - \frac{\|p\|^2}{4\lambda} + p^\top q \big),
\end{aligned}$$

where λ is strictly positive because the constraint $\|p - q\|^2 \leq r^2$ is strictly active at the optimal solution. By letting $\partial H(\lambda)/\partial\lambda = 0$, the optimal λ is written as

$$\lambda^* := \frac{\|p\|}{2r} = \arg\max_{\lambda > 0} \ H(\lambda).$$

Substituting λ^* into $H(\lambda)$,

$$p^\top q - \|p\| r = \max_{\lambda > 0} \ H(\lambda).$$

The upper bound of $p^\top z$ in (13) can be shown similarly. ■

Proof of Lemma 1. From Proposition 6, the optimal solution w_C^* satisfies

$$\left(w_C^* + C\sum_{i\in[n]}\xi_i(w_C^*)\right)^\top (w_C^* - \hat{w}_{\tilde{C}}) \le 0, \tag{14}$$

where $\xi_i(w_C^*)$ is a subgradient of ℓ_i at $w = w_C^*$ for any $i \in [n]$.

Since from ℓ_i is convex for any $i \in [n]$ and the definition of a subgradient, we have the following two inequalities:

$$\ell_i(w_C^*) \ge \ell_i(\hat{w}_{\tilde{C}}) + \xi_i(\hat{w}_{\tilde{C}})^\top (w_C^* - \hat{w}_{\tilde{C}}).$$
$$\ell_i(\hat{w}_{\tilde{C}}) \ge \ell_i(w_C^*) + \xi_i(w_C^*)^\top (\hat{w}_{\tilde{C}} - w_C^*).$$

Combining these two inequalities, we have

$$\xi_i(w_C^*)^\top (w_C^* - \hat{w}_{\tilde{C}}) \ge \xi_i(\hat{w}_{\tilde{C}})^\top (w_C^* - \hat{w}_{\tilde{C}}). \tag{15}$$

Substituting (15) into (14),

$$w_C^{*\top}(w_C^* - \hat{w}_{\tilde{C}}) + C\sum_{i\in[n]}\xi_i(\hat{w}_{\tilde{C}})^\top (w_C^* - \hat{w}_{\tilde{C}}) \le 0. \tag{16}$$

From (4),

$$\sum_{i\in[n]}\xi_i(\hat{w}_{\tilde{C}}) = \frac{1}{\tilde{C}}\Big(g(\hat{w}_{\tilde{C}}) - \hat{w}_{\tilde{C}}\Big). \tag{17}$$

Substituting (17) into (16),

$$\begin{aligned} & w_C^{*\top}(w_C^* - \hat{w}_{\tilde{C}}) + \frac{C}{\tilde{C}}\Big(g(\hat{w}_{\tilde{C}}) - \hat{w}_{\tilde{C}}\Big)^\top (w_C^* - \hat{w}_{\tilde{C}}) \le 0 \\ \Leftrightarrow\ & \left\|w_C^* - \frac{1}{2}\Big(\hat{w} - \frac{C}{\tilde{C}}(g(\hat{w}) - \hat{w})\Big)\right\|^2 \le \left(\frac{1}{2}\left\|\hat{w} + \frac{C}{\tilde{C}}(g(\hat{w}) - \hat{w})\right\|\right)^2. \end{aligned}$$

The lower bound $LB(w_C^{*\top} x_i' | \hat{w}_{\tilde{C}})$ is given by solving the following optimization problem:

$$\min_{w_C^*} \quad w_C^{*\top} x_i' \qquad \text{s.t.} \quad \left\|w_C^* - \frac{1}{2}\Big(\hat{w} - \frac{C}{\tilde{C}}(g(\hat{w}) - \hat{w})\Big)\right\|^2 \le \left(\frac{1}{2}\left\|\hat{w} + \frac{C}{\tilde{C}}(g(\hat{w}) - \hat{w})\right\|\right)^2. \tag{18}$$

Using Proposition 7, the solution of (18) is given as

$$\begin{aligned} LB(w_C^{*\top} x_i' | \hat{w}_{\tilde{C}}) &= \frac{1}{2} x_i'^\top \Big(\hat{w} - \frac{C}{\tilde{C}}(g(\hat{w}) - \hat{w})\Big) - \|x_i'\| \left\|\frac{1}{2}\Big(\hat{w} + \frac{C}{\tilde{C}}(g(\hat{w}) - \hat{w})\Big)\right\| \\ &\le \frac{1}{2} x_i'^\top \Big(\hat{w} - \frac{C}{\tilde{C}}(g(\hat{w}) - \hat{w})\Big) - \frac{1}{2}\|x_i'\| \Big(\Big|1 - \frac{C}{\tilde{C}}\Big|\|\hat{w}\| + \frac{C}{\tilde{C}}\|g(\hat{w})\|\Big) \\ &= \begin{cases} \alpha(\hat{w}_{\tilde{C}}, x_i') - \frac{1}{\tilde{C}}(\beta(\hat{w}_{\tilde{C}}, x_i') + \gamma(g(\hat{w}_{\tilde{C}}), x_i'))C, & \text{if } C \ge \tilde{C}, \\ -\beta(\hat{w}_{\tilde{C}}, x_i') + \frac{1}{\tilde{C}}(\alpha(\hat{w}_{\tilde{C}}, x_i') + \delta(g(\hat{w}_{\tilde{C}}), x_i'))C, & \text{if } C < \tilde{C}. \end{cases} \end{aligned}$$

Similarly, the upper bound $UB(w_C^{*\top} x_i' | \hat{w}_{\tilde{C}})$ is given by solving the following optimization problem

$$\max_{w_C^*} \quad w_C^{*\top} x_i' \qquad \text{s.t.} \quad \left\|w_C^* - \frac{1}{2}\Big(\hat{w} - \frac{C}{\tilde{C}}(g(\hat{w}) - \hat{w})\Big)\right\|^2 \le \left(\frac{1}{2}\left\|\hat{w} + \frac{C}{\tilde{C}}(g(\hat{w}) - \hat{w})\right\|\right)^2, \tag{19}$$

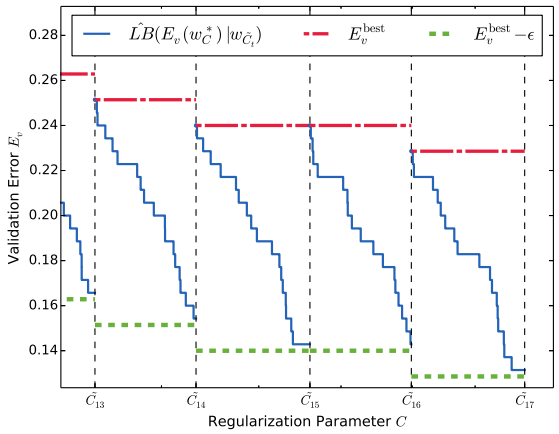


Figure 4: An illustrative example of Algorithm 2 behavior. The blue real lines represent the validation error lower bound. The red chained lines and green dashed lines indicate the current best validation error upper bound E_v^{best} and $E_v^{\text{best}} - \varepsilon$, respectively. If the blue validation error lower bound falls below the green ones, the validation error can be smaller by ε than the current best. In such a case, the algorithm computes the next approximate solution, and update the validation error lower bound based on the new approximate solution. The plot is an enlarged view of the region from $\tilde{C}_{13}$ to $\tilde{C}_{17}$ in Figure 3 (a) in §5.

and the solution of (19) is given as

$$
\begin{aligned}
UB(w_C^{*\top} x_i' | \hat{w}_{\tilde{C}}) &= \frac{1}{2} x_i'^{\top} \Big(\hat{w} - \frac{C}{\tilde{C}} (g(\hat{w}) - \hat{w}) \Big) + \|x_i'\| \Big\| \frac{1}{2} \Big(\hat{w} + \frac{C}{\tilde{C}} (g(\hat{w}) - \hat{w}) \Big) \Big\| \\
&\geq \frac{1}{2} x_i'^{\top} \Big(\hat{w} - \frac{C}{\tilde{C}} (g(\hat{w}) - \hat{w}) \Big) + \frac{1}{2} \|x_i'\| \Big(\Big| 1 - \frac{C}{\tilde{C}} \Big| \|\hat{w}\| + \frac{C}{\tilde{C}} \|g(\hat{w})\| \Big) \\
&= \begin{cases} -\beta(\hat{w}_{\tilde{C}}, x_i') + \frac{1}{\tilde{C}} (\alpha(\hat{w}_{\tilde{C}}, x_i') + \delta(g(\hat{w}_{\tilde{C}}), x_i')) C, & \text{if } C \geq \tilde{C}, \\ \alpha(\hat{w}_{\tilde{C}}, x_i') - \frac{1}{\tilde{C}} (\beta(\hat{w}_{\tilde{C}}, x_i') + \gamma(g(\hat{w}_{\tilde{C}}), x_i')) C, & \text{if } C < \tilde{C}. \end{cases}
\end{aligned}
$$

■

Remark 8. *We note that the idea of using Propositions 6 and 7 for proving Lemma 1 is inspired from recent studies on safe screening [9, 28, 21, 19, 27]. Safe screening has been introduced in the context of sparse modeling. It allows us to identify sparse features or instances before actually solving the optimization problem. A key technique used in those studies is to bound Lagrange multipliers at the optimal solution (Lagrange multiplier values at the optimal solution tell us which features or instances are active or non-active) in somewhat similar way as we did in §3. Our main contribution is to borrow this idea for representing a validation error lower bound as a function of the regularization parameter, and show that it can be used for finding an approximately optimal regularization parameter with theoretical guarantee.*

B Details of the speed-up tricks for finding an ε-approximate regularization parameter

In this appendix, we first describe two modifications of the basic algorithm for finding an ε-approximate regularization parameter presented in §4.2 for further speed-up.

Trick1 The efficiency of the algorithm depends on how far one can step forward in each iteration. We see in (10) that the step size $\tilde{C}_{t+1} - \tilde{C}_t$ is large if the current minimum validation error upper bound E_v^{best} is small. In other words, the step size will be small until we have sufficiently small E_v^{best}. It suggests that, if we can find small enough E_v^{best} at an *earlier* stage of the algorithm, we can reduce the total computational cost of the algorithm. In order to find sufficiently small E_v^{best} as early as possible, we propose a simple heuristic approach, where we first roughly search over the entire range by a rough grid search.

Trick2 Our next modification for speed-up is to use

$$LB(E_v(w_C^*)|\hat{w}_{\tilde{C}_t}, \hat{w}_{\tilde{C}_{t+1}}) := \max\{LB(E_v(w_C^*)|\hat{w}_{\tilde{C}_t}), LB(E_v(w_C^*)|\hat{w}_{\tilde{C}_{t+1}})\},$$

for computing the validation error lower bound in $C \in [\tilde{C}_t, \tilde{C}_{t+1}]$. It provides a tighter validation error lower bounds than using $LB(E_v(w_C^*)|\hat{w}_{\tilde{C}_t})$ alone, meaning that larger step might be allowed in each iteration. However, we cannot actually compute $LB(E_v(w_C^*)|\hat{w}_{\tilde{C}_{t+1}})$ before we fix $\tilde{C}_{t+1}$. We thus propose a simple trial-and-error approach. Specifically, we step forward a little bit further than (10) when we select the next $\tilde{C}_{t+1}$. After we fix $\tilde{C}_{t+1}$, we compute an approximate solution $\hat{w}_{\tilde{C}_{t+1}}$ and then check whether the validation error $E_v(w_C^*)$ is not smaller by ε than the current minimum for $C \in [\tilde{C}_t, \tilde{C}_{t+1}]$ by using now available $LB(E_v(w_C^*)|\hat{w}_{\tilde{C}_t}, \hat{w}_{\tilde{C}_{t+1}})$.

Algorithm 3 is the pseudo-code of the proposed algorithm along with tricks 1 and 2.

There are two additional input parameters $m \in \mathbb{N}$ and $\rho > 1$. The former is used for trick1, where we initially compute m approximate solutions for regularization parameter values evenly allocated in the interval $[C_l, C_u]$ in the logarithmic scale. Trick1 is described at lines 2-9 in Algorithm 3.

The latter $\rho > 1$ is used for trick2, where the next regularization parameter value is determined in trial-and-error manner. To formally describe trick2, let us define a set Γ as a function of w in the following way

$$\Gamma(w_{\tilde{C}}) := \left\{\frac{\beta(w_{\tilde{C}}, x_i')}{\alpha(w_{\tilde{C}}, x_i') + \delta(g(w_{\tilde{C}}), x_i')}\tilde{C}\right\}_{i\in\mathcal{P}} \cup \left\{\frac{\alpha(w_{\tilde{C}}, x_i')}{\beta(w_{\tilde{C}}, x_i') + \gamma(g(w_{\tilde{C}}), x_i')}\tilde{C}\right\}_{i\in\mathcal{N}}.$$

Then, our initial trial step is written as

$$C^{\mathrm{tmp}} := (\lfloor n'(LB(E_v(w_{\tilde{C}_t}^*)|\hat{w}_{\tilde{C}_t}) - E_v^{\mathrm{best}} + \rho\varepsilon)\rfloor + 1)^{\mathrm{th}}(\Gamma(\hat{w}_{\tilde{C}_t})), \tag{20}$$

where $\rho > 1$ represents how far we step forward. We then compute an approximate solution $\hat{w}_{C^{\mathrm{tmp}}}$, and obtain a validation error lower bound $LB(E_v(w_C^*)|\hat{w}_{\tilde{C}_t}, \hat{w}_{\tilde{C}^{\mathrm{tmp}}})$ by combining $LB(E_v(w_C^*)|\hat{w}_{\tilde{C}_t})$ and

$LB(E_v(w_C^*)|\hat{w}_{C^{\rm tmp}})$. For accepting this trial step, we need to make sure that the lower bounds are not smaller by ε than the current best $E_v^{\rm best}$ for any $C \in [C_t, C^{\rm tmp}]$. To this end, we investigate where the two lower bounds $LB(E_v(w_C^*)|\hat{w}_{\tilde{C}_t})$ and $LB(E_v(w_C^*)|\hat{w}_{C^{\rm tmp}})$ go below $E_v^{\rm best} - \varepsilon$. To formulate this, let us define the following two functions

$$C^R(\hat{w}_{C(L)}) := (\lfloor n'(LB(E_v(w^*_{C(L)})|\hat{w}_{C(L)}) \ \ - E_v^{\rm best} + \varepsilon)\rfloor + 1)^{\rm th}(\Gamma(\hat{w}_{C(L)})), \tag{21}$$

$$C^L(\hat{w}_{C(R))}) := (\lfloor n'(LB(E_v(w^*_{C(R)})|\hat{w}_{C(R)}) \ \ - E_v^{\rm best} + \varepsilon)\rfloor + 1)^{\rm TH}(\Delta(\hat{w}_{C(R)})), \tag{22}$$

where, for the latter, we define

$$\Delta(w_{\tilde{C}}) := \left\{ \frac{\alpha(w_{\tilde{C}}, x_i')}{\beta(w_{\tilde{C}}, x_i') + \gamma(g(w_{\tilde{C}}), x_i')} \tilde{C} \right\}_{i \in \mathcal{P}} \cup \left\{ \frac{\beta(w_{\tilde{C}}, x_i')}{\alpha(w_{\tilde{C}}, x_i') + \delta(g(w_{\tilde{C}}), x_i')} \tilde{C} \right\}_{i \in \mathcal{N}},$$

and denote the $k^{\rm TH}$-largest element of Δ as $k^{\rm TH}(\Delta)$ for any natural number k. The trial step to $C^{\rm tmp}$ is accepted if

$$C^L(\hat{w}_{C^{\rm tmp}}) < C^R(\hat{w}_{\tilde{C}_t}).$$

If not, we need to shrink the trial step by using the procedure described in Algorithm 4. Briefly speaking, Algorithm 4 conducts a bisection search until we find two approximate solutions $\hat{w}_{C(L)}$ and $\hat{w}_{C(R)}$ that satisfy $C^L(\hat{w}_{C(L)}) < C^R(\hat{w}_{C(L)})$. We note that, with the use of trick2, the sequence of the regularization parameter values $\tilde{C}_1, \ldots, \tilde{C}_T$ is not necessarily in increasing order because they are computed in trial-and-error manner.

C Approximate regularization path in terms of validation errors

In this appendix, we describe the details of approximate regularization path in terms of validation errors and its experimental results.

By slightly modifying the algorithm, we can compute an ε-approximate *regularization path* whose approximation level is measured in terms of the validation errors. Such an ε-approximate regularization path is formulated as a function

$$W : [C_l, C_u] \to \mathbb{R}^d, C \mapsto w,$$

such that

$$|E_v(W(C)) - E_v(w_C^*)| \leq \varepsilon, \forall C \in [C_l, C_u].$$

In order to compute W, we need an upper bound of the validation errors as well as a lower bound represented as a function of the regularization parameter. Given a solution $\hat{w}_{\tilde{C}}$ for a regularization parameter $\tilde{C}$, our basic idea is to go forward the regularization path as long as the difference between the upper and the lower bounds are not greater than ε. We note that, the approximation quality of our approximate regularization

path is measured in terms of the validation errors, which is more advantageous for hyper-parameter tuning tasks than existing approaches [13, 11, 12, 20] in which the approximation quality is evaluated in terms of the objective function values.

We compute a validation error upper bound based on the following simple facts:

$$y'_i = +1 \text{ and } LB(w_C^{*\top} x'_i | w_{\tilde{C}}) \geq 0 \Rightarrow \text{correctly-classified}, \tag{23a}$$

$$y'_i = -1 \text{ and } UB(w_C^{*\top} x'_i | w_{\tilde{C}}) \leq 0 \Rightarrow \text{correctly-classified}. \tag{23b}$$

Based on these facts, we have a lemma for validation error upper bounds similar to Lemma 3:

Lemma 9. *For a validation instance with $y'_i = +1$, if*

$$\tilde{C} < C \leq \frac{\alpha(\hat{w}_{\tilde{C}}, x'_i)}{\beta(\hat{w}_{\tilde{C}}, x'_i) + \gamma(g(\hat{w}_{\tilde{C}}), x'_i)} \tilde{C} \quad \text{or} \quad \frac{\beta(\hat{w}_{\tilde{C}}, x'_i)}{\alpha(\hat{w}_{\tilde{C}}, x'_i) + \delta(g(\hat{w}_{\tilde{C}}), x'_i)} \tilde{C} \leq C < \tilde{C},$$

then the validation instance (x'_i, y'_i) is correctly-classified. Similarly, for a validation instance with $y'_i = -1$, if

$$\tilde{C} < C \leq \frac{\beta(\hat{w}_{\tilde{C}}, x'_i)}{\alpha(\hat{w}_{\tilde{C}}, x'_i) + \delta(g(\hat{w}_{\tilde{C}}), x'_i)} \tilde{C} \quad \text{or} \quad \frac{\alpha(\hat{w}_{\tilde{C}}, x'_i)}{\beta(\hat{w}_{\tilde{C}}, x'_i) + \gamma(g(\hat{w}_{\tilde{C}}), x'_i)} \tilde{C} \leq C < \tilde{C},$$

then the validation instance (x'_i, y'_i) is correctly-classified.

This lemma can be easily shown by applying (5) to (23).

Using Lemma 9, an upper bound of the validation errors is represented as a function of the regularization parameter C in the following form.

Theorem 10. *Using an approximate solution $\hat{w}_{\tilde{C}}$ for a regularization parameter $\tilde{C}$, the validation error $E_v(w_C^*)$ for any $C > 0$ other than $\tilde{C}$ satisfies*

$$\begin{aligned} E_v(w_C^*) &\leq UB(E_v(w_C^*) | \hat{w}_{\tilde{C}}) \\ &:= 1 - \frac{1}{n'} \Bigg(\sum_{y'_i = +1} I\left(\tilde{C} < C \leq \frac{\alpha(\hat{w}_{\tilde{C}}, x'_i)}{\beta(\hat{w}_{\tilde{C}}, x'_i) + \gamma(g(\hat{w}_{\tilde{C}}), x'_i)} \tilde{C} \right) \\ &\quad + \sum_{y'_i = +1} I\left(\frac{\beta(\hat{w}_{\tilde{C}}, x'_i)}{\alpha(\hat{w}_{\tilde{C}}, x'_i) + \delta(g(\hat{w}_{\tilde{C}}), x'_i)} \tilde{C} \leq C < \tilde{C} \right) \\ &\quad + \sum_{y'_i = -1} I\left(\tilde{C} < C \leq \frac{\beta(\hat{w}_{\tilde{C}}, x'_i)}{\alpha(\hat{w}_{\tilde{C}}, x'_i) + \delta(g(\hat{w}_{\tilde{C}}), x'_i)} \tilde{C} \right) \\ &\quad + \sum_{y'_i = -1} I\left(\frac{\alpha(\hat{w}_{\tilde{C}}, x'_i)}{\beta(\hat{w}_{\tilde{C}}, x'_i) + \gamma(g(\hat{w}_{\tilde{C}}), x'_i)} \tilde{C} \leq C < \tilde{C} \right) \Bigg). \end{aligned} \tag{24}$$

Theorem 10 is a direct consequence of Lemma 9.

C.1 Algorithm

Algorithm 5 is the pseudo-code of our approximate regularization path.

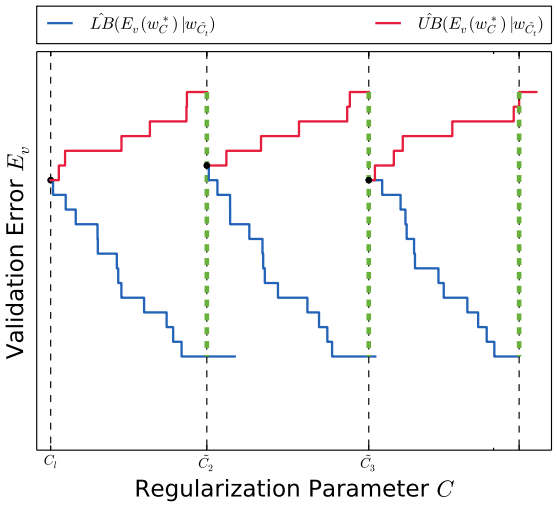


Figure 5: An illustrative image of tacking the approximate regularization path in terms of validation errors algorithm behavior. The blue real lines and red lines represent the validation error lower and upper bounds, respectively. The green dashed lines indicate the difference between the validation error lower and upper bounds. If the green dashed is greater than or equal to ε , we miss tracking ε approximation path. In such a case, the algorithm computes the next approximate solution, and update the validation error lower and upper bounds based on the new approximate solution.

The main difference between Algorithm 2 and Algorithm 5 is in how to determine the next regularization parameter value $\tilde{C}_{t+1}$. For tracking an approximate solution path, we need to find the smallest $\tilde{C}_{t+1} > C$ such that the difference between the upper and the lower bounds $UB(E_v(w_C^*)|\hat{w}_C) - LB(E_v(w_C^*)|\hat{w}_C)$ is greater than or equal to ε. To formulate this, let us define

$$\mathcal{P}' := \{i \in [n'] | y_i' = +1, LB(w_{\tilde{C}_t}^{*\top} x_i' | \hat{w}_{\tilde{C}_t}) \geq 0\}, \mathcal{N}' := \{i \in [n'] | y_i' = -1, UB(w_{\tilde{C}_t}^{*\top} x_i' | \hat{w}_{\tilde{C}_t}) \leq 0\}.$$

and

$$\Lambda(\hat{w}_{\tilde{C}_t}) := \left\{ \frac{\beta(\hat{w}_{\tilde{C}_t}, x_i')}{\alpha(\hat{w}_{\tilde{C}_t}, x_i') + \delta(g(\hat{w}_{\tilde{C}_t}), x_i')} \tilde{C}_t \right\}_{i \in \mathcal{P} \cup \mathcal{N}'} \cup \left\{ \frac{\alpha(\hat{w}_{\tilde{C}_t}, x_i')}{\beta(\hat{w}_{\tilde{C}_t}, x_i') + \gamma(g(\hat{w}_{\tilde{C}_t}), x_i')} \tilde{C}_t \right\}_{i \in \mathcal{N} \cup \mathcal{P}'},$$

Then, $\tilde{C}_{t+1}$ that meets the above requirement is formulated as

$$\tilde{C}_{t+1} \leftarrow (\lfloor n'(LB(E_v(w_{\tilde{C}_t}^*)|\hat{w}_{\tilde{C}_t}) - UB(E_v(w_{\tilde{C}_t}^*)|\hat{w}_{\tilde{C}_t}) + \varepsilon) \rfloor + 1)^{\text{th}}(\Lambda). \tag{25}$$

Figure 5 depicts how C_{t+1} is determined.

Using the output of Algorithm 5 , our approximate regularization path is written as

$$W : [C_l, C_u] \to \mathbb{R}^d, \; C \mapsto \sum_{i=1}^{T} \mathbf{1}_{[C_i, C_{i+1})}(C) \hat{w}_{C_i},$$

where

$$\mathbf{1}_{[C_i, C_{i+1})}(C) = \begin{cases} 1 & \text{if} \quad C \in [C_i, C_{i+1}), \\ 0 & \text{if} \quad C \notin [C_i, C_{i+1}). \end{cases}$$

In approximate regularization path computation, we need a special treatment in a pathological situation that the signs of the scores of multiple validation instances change at one time at a regularization parameter value C. Such a pathological situation is formally stated as follows. Let

$$\Omega := \{i \in [n'] | y'_i = +1, LB(w^{*\top}_{\tilde{C}_t} x'_i | \hat{w}_{\tilde{C}_t}) = 0\} \cup \{i \in [n'] | y'_i = -1, UB(w^{*\top}_{\tilde{C}_t} x'_i | \hat{w}_{\tilde{C}_t}) = 0\}.$$

Then, if the size of Ω is greater than

$$(\lfloor n'(LB(E_v(w^*_{\tilde{C}_t}) | \hat{w}_{\tilde{C}_t}) - UB(E_v(w^*_{\tilde{C}_t}) | \hat{w}_{\tilde{C}_t}) + \varepsilon) \rfloor + 1),$$

Algorithm 5 does not work properly. Although such a pathological situation can be considered as an exceptional case and treated by tedious book-keeping operations, in the following experiments, we simply add an constraint that $\tilde{C}_{t+1} - \tilde{C}_t \geq 10^{-6}$.

C.2 Experiments

Here, we describe the experimental results on approximate regularization path computation. The experimental setup is same as that in §5. Since we cannot use speed-up tricks here, we have two algorithm options. In the first option (**op4**), we used optimal solutions $\{w^*_{\tilde{C}_t}\}_{t \in [T]}$ for computing CV error lower bounds. In the second option (**op5**), we instead used approximate solutions $\{\hat{w}_{\tilde{C}_t}\}_{t \in [T]}$. Table 3 shows the experimental results. Compared with the results in Table 1, we needed to solve more optimization problems (denoted as T) and hence the total computational cost is larger than simply finding an ε-approximate regularization parameter. For large datasets D9 and D10 with $\varepsilon = 0$, we could not finish the computations within 100 hours.

Table 3: Complexities and computational costs of approximate regularization path computation experiments. For each of the three options and $\varepsilon \in \{0.10, 0.05, 0.01, 0\}$, the number of optimization problems solved (denoted as T) and the total computational costs (denoted as time) are listed. Note that, for **op5**, there are no results for $\varepsilon = 0$. For D9 and D10 with $\varepsilon = 0$, we could not finish the computations within 100 hours.

		op4 (using $w^*_{\tilde{C}}$)		**op5** (using $\hat{w}_{\tilde{C}}$)			**op4** (using $w^*_{\tilde{C}}$)		**op5** **(using $\hat{w}_{\tilde{C}}$)**	
ε		T	time (sec)	T	time (sec)		T	time (sec)	T	time (sec)
0.10	D1	91	0.208	96	0.073	D6	238	4.828	240	1.691
0.05		150	0.284	180	0.118		503	9.185	507	3.518
0.01		698	1.063	2095	0.597		2332	31.17	3300	17.16
0		6960	7.983	N.A.			74767	836.7	N.A.	
0.10	D2	504	0.367	510	0.246	D7	732	18.56	742	10.49
0.05		902	0.563	982	0.444		1316	31.77	1385	18.88
0.01		4549	2.711	9404	2.365		5820	118.4	7700	76.80
0		94612	68.31	N.A.			1583578	43212	N.A.	
0.10	D3	175	1.739	186	0.592	D8	227	1.991	229	1.410
0.05		314	2.615	374	1.005		469	3.987	475	2.872
0.01		1329	9.360	3248	3.409		2382	17.95	2385	14.75
0		56123	292.3	N.A.			397801	5481	N.A.	
0.10	D4	84	0.472	86	0.201	D9	352	844.0	357	302.6
0.05		156	0.798	162	0.355		717	1209	725	624.4
0.01		710	2.816	1218	1.497		3741	4985	11631	11185
0		14833	48.06	N.A.			> 100h		N.A.	
0.10	D5	136	0.527	138	0.185	D10	189	145.5	200	45.18
0.05		283	0.936	286	0.368		262	203.7	272	61.07
0.01		1561	3.840	2306	2.086		832	524.7	851	179.7
0		50101	104.9	N.A.			> 100h		N.A.	

D Adaptation to cross-validation setup

All the methods presented above can be straightforwardly adapted to a cross-validation (CV) setup. Consider k-fold CV where n instances are divided into k disjoint subsets $\{F_\kappa\}_{\kappa\in[k]}$ with almost equal size. Let $w(\kappa)^*_C$ be the optimal solution trained without using the instances in F_κ. Then, the k-fold CV error is defined as

$$E_{k\mathrm{CV}}(C) := \frac{1}{n}\sum_{\kappa\in[k]}\sum_{i\in F_\kappa} I\big(y_i w(\kappa)^{*\top}_C x_i < 0\big),$$

where, note that, the CV error is not a function of w, but a function of C. Our algorithm can find an ε-approximate regularization parameter at which the k-fold CV error is guaranteed to be no greater by ε than the smallest possible k-fold CV error. For each of the k folds, we can compute a validation error lower bound as described before. A lower bound of the entire k-fold CV error can be obtained by simply summing them up.

Algorithm 3 : Finding an ε-approximate regularization parameter with approximate solutions using tricks 1 and 2

Input: $\{(x_i, y_i)\}_{i\in[n]}$, $\{(x'_i, y'_i)\}_{i\in[n']}$, C_l, C_u, ε, m, ρ

$C^{\mathrm{best}} \leftarrow C_l$, $E_v^{\mathrm{best}} \leftarrow 1$

$s \leftarrow \frac{\log_{10}(C_u)-\log_{10}(C_l)}{m}$

for $h = 0$ to $m-1$ **do**

  $\bar{C}_h \leftarrow 10^{(\log_{10}(C_l)+h\times s)}$

  $\hat{w}_{\bar{C}_h} \leftarrow$ solve (1) approximately for $C = \bar{C}_h$

  if $UB(E_v(w^*_{\bar{C}_h})|\hat{w}_{\bar{C}_h}) < E_v^{\mathrm{best}}$ **then**

    $E_v^{\mathrm{best}} \leftarrow UB(E_v(w^*_{\bar{C}_h})|\hat{w}_{\bar{C}_h})$, $C^{\mathrm{best}} \leftarrow \bar{C}_h$

  end if

end for

$\bar{C}_m \leftarrow C_u$, $t \leftarrow 1$

for $h = 0$ to $m-1$ **do**

  $\tilde{C}_t \leftarrow \bar{C}_h$, $\hat{w}_{\tilde{C}_t} \leftarrow \hat{w}_{\bar{C}_h}$

  while $\tilde{C}_t \leq \bar{C}_{h+1}$ **do**

    Set C^{tmp} by (20) using $\hat{w}_{\tilde{C}_t}$

    if $C^{\mathrm{tmp}} > \bar{C}_{h+1}$ **then**

      Set C^{tmp} by (22) using $\hat{w}_{\tilde{C}_t}$

      if $C^{\mathrm{tmp}} > \bar{C}_{h+1}$ **then**

        break while loop

      end if

    end if

    $\hat{w}_{C^{\mathrm{tmp}}} \leftarrow$ solve (1) approximately for $C = C^{\mathrm{tmp}}$

    Compute $UB(E_v(w^*_{C^{\mathrm{tmp}}})|\hat{w}_{C^{\mathrm{tmp}}})$ by (8b).

    if $UB(E_v(w^*_{C^{\mathrm{tmp}}})|\hat{w}_{C^{\mathrm{tmp}}}) < E_v^{\mathrm{best}}$ **then**

      $E_v^{\mathrm{best}} \leftarrow UB(E_v(w^*_{C^{\mathrm{tmp}}})|\hat{w}_{C^{\mathrm{tmp}}})$

      $C^{\mathrm{best}} \leftarrow C^{\mathrm{tmp}}$

    end if

    $r \leftarrow 0$

    RecursiveCheck($\tilde{C}_t, C^{\mathrm{tmp}}, \hat{w}_{\tilde{C}_t}, \hat{w}_{C^{\mathrm{tmp}}}, r$)

    $\tilde{C}_{t+r+1} \leftarrow C^{\mathrm{tmp}}$, $\hat{w}_{\tilde{C}_{t+r+1}} \leftarrow \hat{w}_{C^{\mathrm{tmp}}}$

    $t \leftarrow t + r + 1$

  end while

end for

Output: $C^{\mathrm{best}} \in \mathcal{C}(\varepsilon)$.

Algorithm 4 : RecursiveCheck $(C(L), C(R), \hat{w}_{C(L)}, \hat{w}_{C(R)}, r)$

Compute $C^R(\hat{w}_{C(L)})$ in (21).

Compute $C^L(\hat{w}_{C(R)})$ in (22).

if $C^L(\hat{w}_{C(R)}) < C^R(\hat{w}_{C(L)})$ **then**

  return

else

  $r \leftarrow r + 1$

  $\tilde{C}_{t+r} \leftarrow \frac{1}{2}(C^L(\hat{w}_{C(R)}) + C^R(\hat{w}_{C(L)}))$

  $\hat{w}_{\tilde{C}_{t+r}} \leftarrow$ solve (1) approximately for $C = \tilde{C}_{t+r}$

  if $UB(E_v(w^*_{\tilde{C}_{t+r}})|\hat{w}_{\tilde{C}_{t+r}}) < E_v^{\text{best}}$ **then**

    $E_v^{\text{best}} \leftarrow UB(E_v(w^*_{\tilde{C}_{t+r}})|\hat{w}_{\tilde{C}_{t+r}})$

    $C^{\text{best}} \leftarrow \tilde{C}_{t+r}$

  end if

  RecursiveCheck$(C(L), \tilde{C}_{t+r}, \hat{w}_{C(L)}, \hat{w}_{\tilde{C}_{t+r}}, r)$

  RecursiveCheck$(\tilde{C}_{t+r}, C(R), \hat{w}_{\tilde{C}_{t+r}}, \hat{w}_{C(R)}, r)$

end if

Algorithm 5 : Tracking an ε-Approximate Regularization Path

Input: $\{(x_i, y_i)\}_{i\in[n]}$, $\{(x'_i, y'_i)\}_{i\in[n']}$, C_l, C_u, ε

1: $t \leftarrow 1$, $\tilde{C}_t \leftarrow C_l$

2: **while** $\tilde{C}_t \leq C_u$ **do**

3: Solve (1) approximately at $C = \tilde{C}_t$ and obtain $\hat{w}_{\tilde{C}_t}$

4: Set $\tilde{C}_{t+1}$ by (25)

5: $t \leftarrow t + 1$

6: **end while**

7: $T \leftarrow t - 1$

Output: $C_1, \ldots, C_{T+1}$, $\hat{w}_{C_1}, \ldots, \hat{w}_{C_T}$

References

[1] J. Bergstra and Y. Bengio. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13:281–305, 2012.

[2] P D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[3] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[4] C. Chang and C. Lin. LIBSVM : A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, 2:1–39, 2011.

[5] O. Chapelle. Training a support vector machine in the primal. *Neural computation*, 19:1155–1178, 2007.

[6] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46:131–159, 2002.

[7] K. Chung, W. Kao, C. Sun, L. Wang, and C. Lin. Radius margin bounds for support vector machines with the RBF kernel. *Neural computation*, 2003.

[8] B. Efron, T. Hastie, I. Johnstone, and R. TIbshirani. Least angle regression. *Annals of Statistics*, 32(2):407–499, 2004.

[9] L. El Ghaoui, V. Viallon, and T. Rabbani. Safe feature elimination in sparse supervised learning. *Pacific Journal of Optimization*, 2012.

[10] R. Fan, K. Chang, and C. Hsieh. LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning*, 9:1871–1874, 2008.

[11] J. Giesen, M. Jaggi, and S. Laue. Approximating Parameterized Convex Optimization Problems. *ACM Transactions on Algorithms*, 9, 2012.

[12] J. Giesen, S. Laue, and Wieschollek P. Robust and Efficient Kernel Hyperparameter Paths with Guarantees. In *International Conference on Machine Learning*, 2014.

[13] J. Giesen, J. Mueller, S. Laue, and S. Swiercy. Approximating Concavely Parameterized Optimization Problems. In *Advances in Neural Information Processing Systems*, 2012.

[14] G. Guennebaud, B. Jacob, and Others. Eigen v3. http://eigen.tuxfamily.org, 2010.

[15] T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5:1391–1415, 2004.

[16] T. Joachims. Estimating the generalization performance of a SVM efficiently. In *International Conference on Machine Learning*, 2000.

[17] M. Lee, S. Keerthi, C. Ong, and D. DeCoste. An efficient method for computing leave-one-out error in support vector machines with Gaussian kernels. *IEEE Transactions on Neural Networks*, 15:750–7, 2004.

[18] C. Lin, R. Weng, and S. Keerthi. Trust Region Newton Method for Large-Scale Logistic Regression. *The Journal of Machine Learning Research*, 9:627–650, 2008.

[19] J. Liu, Z. Zhao, J. Wang, and J. Ye. Safe Screening with Variational Inequalities and Its Application to Lasso. In *International Conference on Machine Learning*, volume 32, 2014.

[20] J. Mairal and B. Yu. Complexity analysis of the Lasso reguralization path. In *International Conference on Machine Learning*, 2012.

[21] K. Ogawa, Y. Suzuki, and I. Takeuchi. Safe screening of non-support vectors in pathwise SVM computation. In *International Conference on Machine Learning*, 2013.

[22] S. Rosset and J. Zhu. Piecewise linear regularized solution paths. *Annals of Statistics*, 35:1012–1030, 2007.

[23] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning*. Cambridge University Press, 2014.

[24] J. Snoek, H. Larochelle, and R. Adams. Practical Bayesian Optimization of Machine Learning Algorithms. In *Advances in Neural Information Processing Sysrtems*, 2012.

[25] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1996.

[26] V. Vapnik and O. Chapelle. Bounds on Error Expectation for Support Vector Machines. *Neural Computation*, 12:2013–2036, 2000.

[27] J. Wang, J. Zhou, J. Liu, P. Wonka, and J. Ye. A Safe Screening Rule for Sparse Logistic Regression. In *Advances in Neural Information Processing Sysrtems*, 2014.

[28] Z. Xiang, H. Xu, and P. Ramadge. Learning sparse representations of high dimensional data on large scale dictionaries. In *Advances in Neural Information Processing Sysrtems*, 2011.